December 15, 1983

FRA-TM-149

The DIF3D Transport Extension for Discrete Ordinance Neutronics Calculations in Two Dimensions*

by

E. E. Lewis**

Applied Physics Division
Argonne National Laboratory
Argonne, Illinois 60439

FRA TECHNICAL MEMORANDUM NO. 149

*Work supported by the U.S. Department of Energy.

**Permanent address: Department of Mechanical and Nuclear Engineering, Northwestern University, Evanston, Illinois, 60201.

DISCLAIMER

Neutronics Calculations in Two Dimensions*

by

E. Lewis

Applied Physics Division
Argonne National Laboratory
Argonne, IL 60439

ABSTRACT

The two-dimensional discrete-ordinates extension of the Argonne National Laboratory neutron diffusion code DIF3D is described. The discrete-ordinates extension is incorporated into x-y, r-z and triangular geometries. Only those sections of DIF3D dealing with the within-group scalar flux calculations are modified. The extension includes two features which facilitate the conversion from diffusion to discrete ordinate computations. An isotropic transport correction insures that results are compatible with anisotropic diffusion theory, and the use of two-cyclic overrelaxation allows the retention of the diffusion iteration and acceleration techniques in the discrete ordinates extension. Derivation of the discrete ordinates equation is included for each geometry along with discussions of the solution algorithms and of other program information relevant to the discrete ordinates extensions.

*Work supported by the U.S. Department of Energy.

TABLE OF CONTENTS

LIST OF FIGURES

The DIF3D Transport Extension for Discrete Ordinates Neutronics Calculations in Two Dimensions

I. INTRODUCTION

Multigroup diffusion theory is the most widely used transport approximation for performing fast reactor neutronics analysis. At Argonne National Laboratory such calculations are performed in one, two or three dimensions using the DIF3D code.[1,2] In a variety of circumstances, however, accuracy requires that diffusion theory be replaced by a higher-order transport method such as the discrete ordinates or S_N method. This may occur, for example, in the analysis of the flux gradients in heterogeneous cores, and in the analysis of control rod behavior.

To serve this need a two-dimensional transport extension to DIF3D has been written. In doing this care has been taken to minimize the number of changes that a user must make in converting the input files from diffusion to transport calculations. In particular, the same cross sections and diffusion coefficients are used in the diffusion and transport calculations. In most cases the user need only insert one additional piece of information: the S_N order.

Consistent with this objective, only the within-group calculations of DIF3D are modified. The transport extension uses the group source and boundary conditions from DIF3D, replaces the diffusion with a S_N calculation of the within-group flux, and returns the group scalar flux to DIF3D. Thus the outer iteration strategy and the data handling required for the group sources are not touched. Likewise all of the necessary cross sections for the within-group transport equation are obtained from the diffusion coefficients and removal cross section data by making the assumption of isotropic scattering in the laboratory system. In the case that anisotropic diffusion coefficients are supplied, the transport equation is modified by an angular multiplier[3,4] to yield results that are consistent with anisotropic diffusion theory in the limit that P_1 theory is valid.

The transport and diffusion versions of DIF3D use the same acceleration parameters and convergence criteria. This is accomplished by retaining the two-cyclic optimized overrelaxation technique used in the diffusion calculations for the transport extension. This is possible through the use of an up-down iteration scheme[5] for the within-group calculations.

The two-dimensional S_N extension for DIF3D is available in x-y, r-z and triangular geometries. It uses standard S_N quadrature sets that are included in the code for S_2 through S_{16} for the x-y and r-z geometries. In the triangular geometry option six-fold symmetric S_2 through S_8 quadratures are included.

In the following sections, we first state the forms of the within-group transport equations that are used. A brief discussion is given of the unique anisotropic transport technique that is included in the code, and the discrete ordinates approximation is introduced. After a discussion of the iterative

solution procedures, the details of the spatial differencing methods are presented. Computational results are given for a number of benchmark problems, and the report is concluded with user information on input and programming.

II. DERIVATION OF TRANSPORT EQUATIONS

II.1. The Within-Group Transport Equation

Given a group source $Q(\vec{r})$ and some combination of vacuum and reflected boundary conditions, the transport equation may be written as[b]

$$\hat{\Omega}\cdot\vec{\nabla}\,\Psi(\vec{r},\hat{\Omega}) + \Sigma(\vec{r})\Psi(\vec{r},\hat{\Omega}) = \Sigma_s(\vec{r})\phi(\vec{r}) + Q(\vec{r}), \tag{1}$$

where Ψ is the angular flux, and the scalar flux is given by

$$\phi(\vec{r}) = \int d\Omega\Psi(\vec{r},\hat{\Omega}). \tag{2}$$

In what follows we use the normalization

$$\int d\Omega = 1. \tag{3}$$

The boundary conditions are given by

$$\Psi(\vec{r},\hat{\Omega}) = 0; \qquad \hat{\Omega}\cdot\hat{n}<0,\ \vec{r}\ \varepsilon\ \Gamma_v, \tag{4}$$

on the vacuum boundaries, where $\hat{n}$ is the outward normal, and

$$\Psi(\vec{r},\hat{\Omega}) = \Psi(\vec{r},\hat{\Omega}') \qquad \vec{r}\ \varepsilon\ \Gamma_r \tag{5}$$

on the reflected boundaries, where $\hat{\Omega}$ is the angle of reflection corresponding to the incident angle $\hat{\Omega}'$. Since no confusion should result, we have deleted the group index.

The within-group scattering and total cross sections, Σ and Σ_s, are determined from the standard relationships for the removal cross section

$$\Sigma_r = \Sigma - \Sigma_s \tag{6}$$

and the diffusion coefficient

$$D = \frac{1}{3\Sigma} . \tag{7}$$

Inverting these expressions, we obtain

$$\Sigma = \frac{1}{3D} \tag{8}$$

and

$$\Sigma_s = \frac{1}{3D} - \Sigma_r . \tag{9}$$

Consistent with the use in diffusion theory, a buckling approximation to the transverse direction leakage may be included by adding a DB^2 term to the total cross section in Eq. (1).

In the case that anisotropic diffusion coefficients have been provided, the situation is somewhat more complicated. These coefficients arise from the homogenization of slab or pin lattices. Since in both the transport and diffusion version of DIF3D homogenized cross sections are used, the anisotropic homogenization appears as an angular multiplier in the transport equation. The theory of these multipliers has been worked out elsewhere.[3,4] The transport term in Eq. (1) is multiplied by an angular multiplier to yield

$$g(\mu)\hat{\Omega}\cdot\vec{\nabla}\Psi(\vec{r},\hat{\Omega}) + \Sigma(\vec{r})\Psi(\vec{r},\hat{\Omega}) = \Sigma_s(\vec{r})\phi(\vec{r}) + Q(\vec{r}), \tag{10}$$

where $g(\mu)$ is chosen such that if a P_1 approximation is applied to Eq. (10) a diffusion equation will result with the Benoist definition of the anisotropic diffusion coefficients.

In x-y geometry, with a slab lattice, μ is the direction cosine of $\hat{\Omega}$ with respect to the direction perpendicular to the slab. In r-z geometry μ is the direction cosine of $\hat{\Omega}$ with respect to the z axis. No anisotropic correction exists in the hexagonal option of the transport code. In the x-y option the slabs must be parallel to the z axis. In the r-z option slabs parallel or perpendicular to the z axis can be treated, as well as pins perpendicular to the z axis. In x-y geometry pins or slabs perpendicular to the z axis would appear as isotropic transport in the x-y plane. We shall return to these points after the discrete ordinates approximation has been applied to Eq. (10). Then, the multiplier $g(\mu)$ is expressed in terms of the anisotropic diffusion coefficients.

II.2. The Discrete Ordinates Approximation

Formulation of Equations

To obtain a discrete ordinates approximation, Eq. (10) is evaluated at each of a number of values $\hat{\Omega}_n$ of the angular variables:

$$g_n \hat{\Omega}_n \cdot \vec{\nabla} \Psi_n(\vec{r}) + \Sigma(\vec{r}) \Psi_n(\vec{r}) = \Sigma_s(\vec{r}) \phi(\vec{r}) + Q(\vec{r}), \tag{11}$$

where

$$\Psi_n(\vec{r}) \approx \Psi(\vec{r}, \hat{\Omega}_n). \tag{12}$$

The scalar flux is approximated by a quadrature formula taken over the discrete ordinates:

$$\phi(\vec{r}) = \sum_n w_n \Psi_n(\vec{r}), \tag{13}$$

where the w_n are the quadrature weights corresponding to the directions $\hat{\Omega}_n$. Likewise, Eq. (4) and (5) are replaced by

$$\Psi_n(\vec{r}) = 0; \qquad \hat{\Omega}_n \cdot \hat{n} < 0, \quad \vec{r} \in \Gamma_v \tag{14}$$

and

$$\Psi_n(\vec{r}) = \Psi_{n'}(\vec{r}); \qquad \vec{r} \in \Gamma_r, \tag{15}$$

where $\hat{\Omega}_n$ is the reflective direction corresponding to the incident direction $\hat{\Omega}_{n'}$.

Determination of Discrete Ordinates

For two-dimensional transport, the $\hat{\Omega}_n$ are chosen over four of the eight octants of the unit sphere defined by $\hat{\Omega}$. We consider the orthogonal coordinate systems (i.e. x-y and r-z) first. For a S_N approximation, a symmetric quadrature set is chosen in each octant as follows.[7] A set of N/2 positive direction cosines is generated in a manner such that there are (N+2)*N/8 per octant, each with the N/2 values of the direction cosines with respect to each of the three axes. This is shown for the S_8 approximation in Fig. 1. Thus for two-dimensional geometries there are (N+2)*N/2 directions. These possess mirror symmetry about the three orthogonal planes. Thus for reflected boundaries, if the angle of incidence is in the quadrature set, so also will be the angle of reflection. As shown elsewhere[7], once the first direction cosine is chosen, the remaining ones are determined by the symmetry conditions. Likewise, the requirement that the quadrature weights be invariant for rotations about any of the axes imposes symmetry conditions on the w_n. In the DIF3D x-y and r-z transport options, the quadrature formulae are the ones used in the TWOTRAN[8] and TPT[9] codes. The quadrature weights and ordinates are reproduced in Appendix A.

In the triangular option, the unit sphere is divided into twelve parts, and in two dimensions the discrete ordinates are defined over six of these.

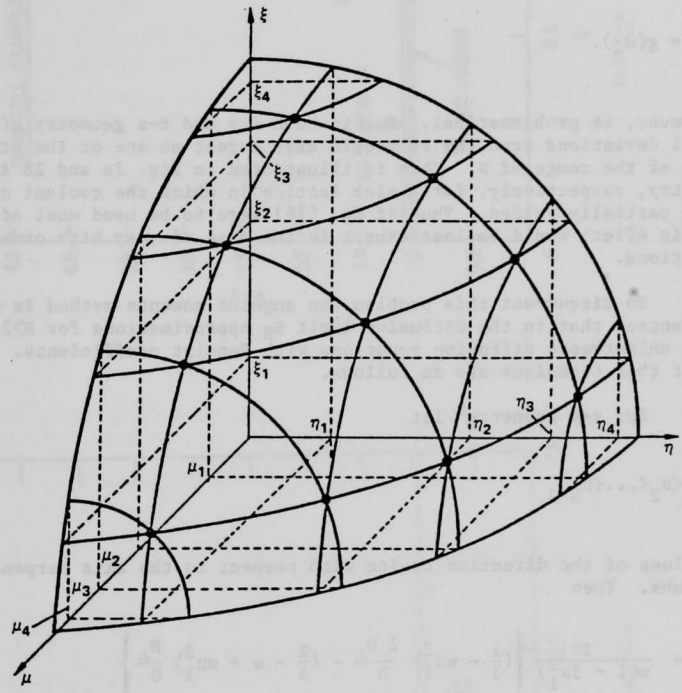


Fig. 1. Level Symmetric Ordinate Set

Within each such sector of the hemisphere there are (N+2)*N/8; discrete ordinates chosen to have mirror symmetry across the three planes on which reflected boundaries may be imposed. Thus in all there are (N+2)*3*N/4 directions for which the transport equation must be solved for two-dimensional triangular problems. In DIF3D the quadrature sets from the DIAMANT2[10] code are used.

Treatment of Anisotropic Transport

The most obvious way to evaluate the angular multipliers in Eq. (11) is to set

$$g_n = g(\hat{\Omega}_n). \tag{16}$$

This, however, is problematical. For in both x-y and r-z geometry $g(\mu)$ has only small deviations from the isotropic case except at one or the other endpoints of the range of μ. This is illustrated in Fig. 2a and 2b in x-y and r-z geometry, respectively, for a slab lattice in which the coolant channels have been partially voided. Thus if Eq. (16) were to be used most of the anisotropic effect would be lost except in the case of very high order S_N approximations.

To circumvent this problem, an angular moments method is used[3,4] that guarantees that in the diffusion limit S_N approximations for N>2 will yield the anisotropic diffusion equations with Benoist coefficients. The results of this technique are as follows.

For x-y geometry[3] let

$$0<\mu_1<\mu_2<\ldots<\mu_{N/2} \tag{17}$$

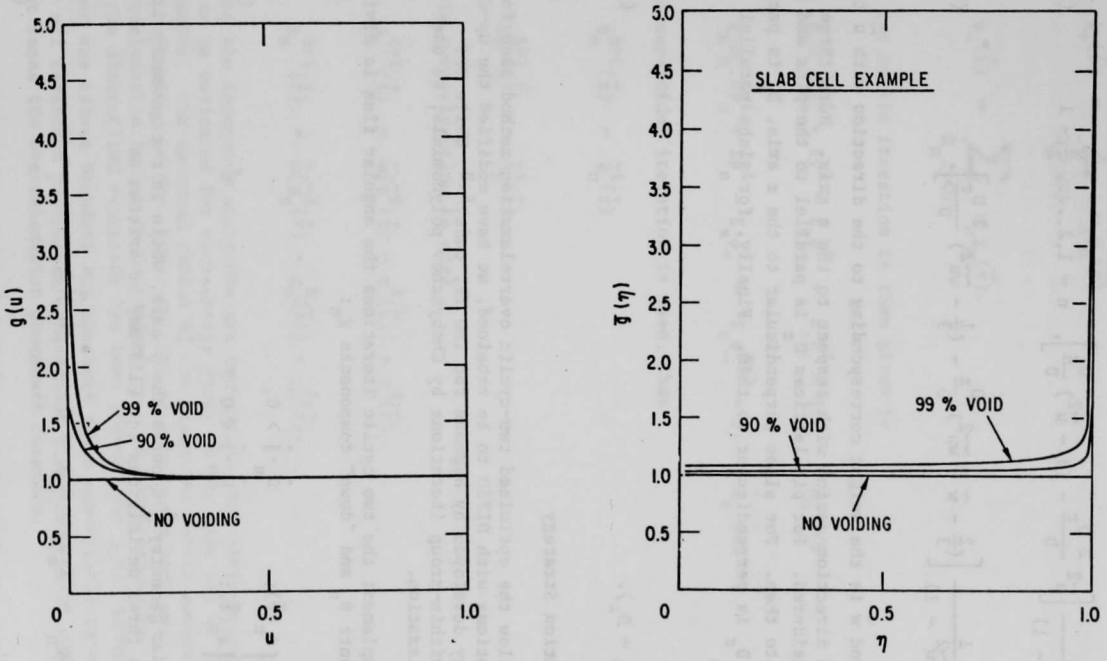
be the values of the direction cosine with respect to the axis perpendicular to the slabs. Then

$$g_1^2 = \frac{1}{w(1 - 3\mu_1^2)} \left\{ \left(\frac{1}{3} - w\mu_1^2\right) \frac{2\,D_{\parallel}}{D} - \left(\frac{2}{3} - w + w\mu_1^2\right) \frac{D_{\perp}}{D} \right\} \tag{18}$$

and

$$g_n^2 = \frac{1}{(1 - 3\mu_1^2)} \left\{ (1 - \mu_1^2) \frac{D_{\perp}}{D} - 2\mu_1^2 \frac{D_{\parallel}}{D} \right\} \qquad n = 2,3,\ldots,N/2. \tag{19}$$

In these expressions $D_{\parallel}$ and $D_{\perp}$ are the diffusion coefficients parallel and perpendicular to the slabs, respectively, and D is defined as in Eq. (7). The weight w is the sum of all the weights for the directions having values of $\hat{\Omega}_n \cdot \hat{i} = \mu_1$.



(a) x-y

(b) r-z

Fig. 2. Multipliers for Anisotropic Transport

For r-z geometry[4] μ is taken as the direction cosine with respect to the z axis, again ordered as in Eq. (17). The multipliers are now given by

$$g_n^2 = \frac{1}{(3\tilde{\mu}^2 - 1)} \left[\tilde{\mu}^2 \frac{2\,D_r}{D} - (1 - \tilde{\mu}^2) \frac{D_z}{D} \right], \quad n = 1,2,\ldots, \frac{N}{2} - 1 \tag{20}$$

and

$$g_{N/2}^2 = \frac{1}{\tilde{w}(3\tilde{w}^2 - 1)} \left[(\frac{2}{3} - \tilde{w} - \tilde{\tilde{w}}\tilde{\mu}^2) \frac{D_z}{D} - (\frac{1}{3} - \tilde{\tilde{w}}\tilde{\mu}^2) \frac{2\,D_r}{D} \right]. \tag{21}$$

where $\tilde{\mu} = \mu_{N/2}$, and $\tilde{w}$ is the weight corresponding to the direction with $\tilde{\mu}$ the largest magnitude direction cosine with respect to the z axis. Here three cases must be considered. For pin lattices D_z is parallel to the pins and D_r is perpendicular to them. For slabs perpendicular to the z axis, D_r is parallel to the slabs and D_z is perpendicular to them. Finally, for slabs parallel to the z axis $D_z = D$

and

$$D_r = \frac{1}{2}(D_{\parallel} + D_{\perp}). \tag{22}$$

II.3. Iteration Strategy

To allow the optimized two-cyclic overrelaxation method used for diffusion calculations with DIF3D to be retained, we have modified the up-down iteration strategy developed by Hageman for the TPT code.[5,9] While TPT accelerates the within-group iterations by Chebyschev polynomials, we use optimized overrelaxation.

To implement the two cyclic iterations the angular flux is divided into "up" components ψ_n and "down" components χ_n:

$$\Psi_n(\vec{r},\hat{\Omega}) = \begin{cases} \psi_n(\vec{r}); & \hat{\Omega}_n \cdot \hat{j} > 0, \\ \chi_n(\vec{r}); & \hat{\Omega}_n \cdot \hat{j} < 0 \end{cases} \tag{23}$$

In x-y or triangular geometry $\hat{j}$ denotes the y axis, while in r-z geometry it is the z axis. With three definitions Eq. (11) may be written as

$$g_n \hat{\Omega}_n \cdot \vec{\nabla} \psi_n + \Sigma \psi_n = \Sigma_s \phi_+ + \Sigma_s \phi_- + Q; \qquad \hat{\Omega} \cdot \hat{j} > 0$$

and (24)

$$g_n \hat{\Omega}_n \cdot \vec{\nabla} \chi_n + \Sigma \chi_n = \Sigma_s \phi_+ + \Sigma_s \phi_- + Q; \qquad \hat{\Omega} \cdot \hat{j} < 0,$$

where the upward and downward components of the scalar flux are defined as

$$\phi_+(\vec{r}) = \sum_{\hat{\Omega}_n \cdot \hat{j} > 0} w_n \psi_n(\vec{r})$$

and (25)

$$\phi_-(\vec{r}) = \sum_{\hat{\Omega}_n \cdot \hat{j} < 0} w_n \chi_n(\vec{r}).$$

The two cyclic iteration is then given by

$$g_n \hat{\Omega}_n \cdot \vec{\nabla} \tilde{\chi}_n^{\ell} + \Sigma \tilde{\chi}_n^{\ell} - \Sigma_s \tilde{\phi}_-^{\ell} = \Sigma_s \phi_+^{\ell} + Q; \qquad \hat{\Omega}_n \cdot \hat{j} < 0$$

(26)

$$g_n \hat{\Omega}_n \cdot \vec{\nabla} \tilde{\psi}_n^{\ell} + \Sigma \tilde{\psi}_n^{\ell} - \Sigma_s \tilde{\phi}_+^{\ell} = \Sigma_s \phi_-^{\ell+1} + Q; \qquad \hat{\Omega}_n \cdot \hat{j} < 0$$

If Gauss-Seidel iteration is used, then

$$\psi_n^{\ell+1}(\vec{r}) = \tilde{\psi}_n^{\ell}(\vec{r})$$

and (27)

$$\chi_n^{\ell+1}(\vec{r}) = \tilde{\chi}_n^{\ell}(\vec{r}),$$

while overrelaxation is used

$$\psi_n^{\ell+1}(\vec{r}) = \tilde{\omega}[\tilde{\psi}_n^{\ell}(\vec{r}) - \psi_n^{\ell}(\vec{r})] + \psi_n^{\ell}(\vec{r})$$

and (28)

$$\chi_n^{\ell+1}(\vec{r}) = \tilde{\omega}[\tilde{\chi}_n^{\ell}(\vec{r}) - \chi_n^{\ell}(\vec{r})] + \chi_n^{\ell}(\vec{r}).$$

Since the foregoing equations are two-cyclic, the optimum overrelaxation factor w can be estimated for each energy group in the same way as for the diffusion equation. The spectral radius of the iteration matrix resulting for the spatial discretization of Eqs. (26) and (28) is determined with w=1, and the optimum overrelaxation factor is determined by a standard procedure. The convergence of the discretized equations has been proven (for g_n=1) provided there is at least one vacuum boundary and has been shown numerically to converge for $g_n \neq 1$. For all reflected boundaries Gaus-Seidel iterations ($\tilde{\omega}$=1) must be used for in most cases the overrelaxation procedure diverges.

In the spatially differential forms of Eq. (26) one iteration consists of a down sweep in which $\tilde{\chi}_n^{\ell}$ is determined for all x (or r) for decreasing values of y (or z). If there are reflected boundary conditions along the lower boundary

$$\psi_n(\vec{r}) = \chi_n(\vec{r}) \qquad \vec{r} \in \Gamma_r \tag{29}$$

in order to initiate the upsweep. Equation (28) is applied to the χ_n before initiating the upsweep. The upsweep is carried out to determine the values of $\psi_n(\vec{r})$ for all x(or r) successively increasing values of y(or z). Equation (28) is applied to overrelax ψ_n, and the boundary values are saved to initiate the next downsweep.

At each level of y(or z) equations of the form

$$g_n \hat{\Omega}_n \cdot \vec{\nabla} \tilde{\psi}_n^{\ell} + \Sigma \tilde{\psi}_n^{\ell} - \Sigma_s \tilde{\phi}_+^{\ell} = S_+ \tag{30}$$

and

$$g_n \hat{\Omega}_n \cdot \vec{\nabla} \tilde{\chi}_n^{\ell} + \Sigma \tilde{\chi}_n^{\ell} - \Sigma_s \tilde{\phi}_-^{\ell} = S_- \tag{31}$$

must be solved, where

$$S_- = \Sigma_s \phi_+^{\ell} + Q \tag{32}$$

and

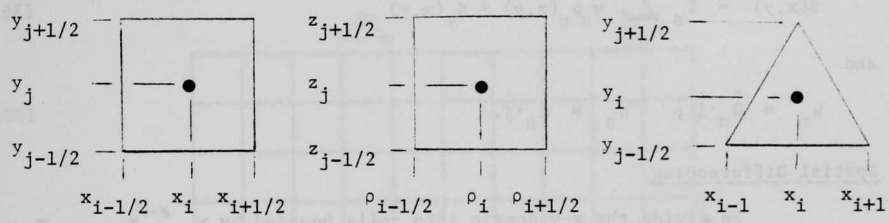
$$S_+ = \tilde{\Sigma}_s \phi_-^{\ell+1} + Q \tag{33}$$

are known from the previous up or down sweep. To solve these equations an inner-inner or line iteration is used to determine the values of $\tilde{\psi}_n^{\ell}$ and $\tilde{\chi}_n^{\ell}$ along a line of fixed y(or z). Before this inner level of iteration can be discussed however, the spatial differencing must be applied to the transport equation. Differencing requires that each of the three geometries be treated separately. However, a few general remarks concerning the grid structures are in order.

The transport extension, like the DIF3D diffusion code, utilizes cell centered grids; the angular and scalar flux values are calculated and stored at the center of x-y, r-z or triangular cells, each with uniform cross sections. Typical cells for the three geometries are shown in Fig. 3 in the r-z cell center line is denoted by $r_{1/2} = 0$. The cross sections are piecewise constant with changes allowed only along cell boundaries (i.e. at the half integer values in x-y and r-z geometry and along the triangle sides in hexagonal geometry).

Since the structure of Eqs. (26) are so similar to one another, we develop only the difference equations for the upsweep $\tilde{\psi}_n$. We delete the iteration superscript for brevity and write

$$g_n \hat{\Omega}_n \cdot \vec{\nabla} \psi_n + \Sigma \psi_n - \Sigma_s \phi_+ = S_+ \tag{34}$$



(a) x-y (b) r-z (c) Triangular

Fig. 3. Grid Cells

as our starting point for each of the three geometries. In all three geometries diamond differencing[7] is used without negative flux fixup.

II.4. x-y Difference Equations

The discrete ordinates equations for ψ_n in x-y geometry are

$$[g_n\mu_n \frac{\partial}{\partial x} + g_n\eta_n \frac{\partial}{\partial y} + \Sigma(x,y)]\psi_n(x,y) = S(x,y), \tag{35}$$

where

$$S(x,y) = \Sigma_s \sum_{\eta_n>0} w_n\psi_n(x,y) + S_+(x,y) \tag{36}$$

and

$$\mu_n = \hat{\Omega}_n\cdot\hat{i} \; ; \quad \eta_n = \hat{\Omega}_n\cdot\hat{j}. \tag{37}$$

Spatial Differencing

We divide the x-y domain into cells bounded by $x_{1/2}, x_{3/2},\cdots x_{I+1/2}$ and $y_{1/2}, y_{3/2},\cdots,y_{J+1/2}$ as shown in Fig. 4. The cross sections are taken to be piecewise constant, and can change values only at the half-integer boundaries in x and y. Within the cell they have the values Σ_{ij}, Σ_{sij}, for example.

To derive difference relations for the i,j cell we integrate Eq. (35) over $x_{i-1/2} < x < x_{i+1/2}$ and $y_{j-1/2} < y < y_{j+1/2}$. We have

$$g_n\mu_n\int_j dy[\psi_n(x_{i+1/2},y) - \psi_n(x_{i-1/2},y)] + g_n\eta_n\int_i dx[\psi_n(x,y_{j+1/2}) - \psi_n(x,y_{j-1/2})]$$

$$+ \Sigma_{ij}\int_i dx\int_j dy\psi_n(x,y) = \int_i dx \int_j dy\, S(x,y), \tag{38}$$

where we have used the abbreviations

$$\int_i dx \equiv \int_{x_{i-1/2}}^{x_{i+1/2}} dx \quad ; \quad \int_j dy \equiv \int_{y_{j-1/2}}^{y_{j+1/2}} dy \; . \tag{39}$$

We further simplify notation by denoting

$$\Delta x_i = x_{i+1/2} - x_{i-1/2} \tag{40}$$

$$\Delta y_j = y_{j+1/2} - y_{j-1/2} \; . \tag{41}$$

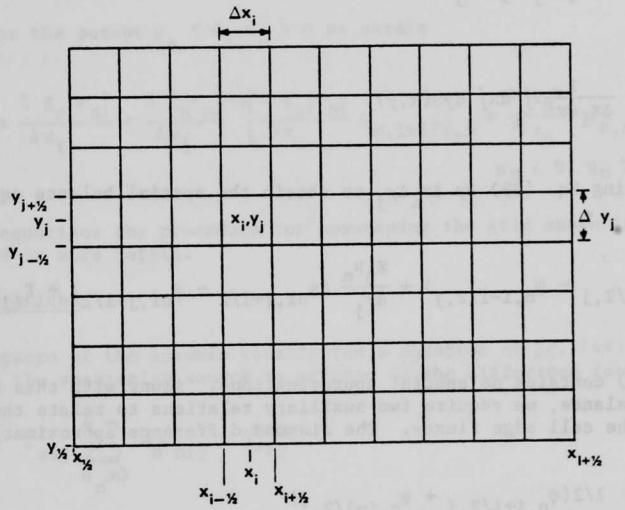


Fig. 4. Grid in x-y Geometry

Then we may define the fluxes averaged over the cell edges and area respectively as

$$\psi_{n,i+1/2,j} = \frac{1}{\Delta y_j} \int dy \psi_n(x_{i+1/2},y) \tag{42}$$

$$\psi_{n,i,j+1/2} = \frac{1}{\Delta x_i} \int_i dx \psi_n(x,y_{j+1/2}) \tag{43}$$

$$\psi_{nij} = \frac{1}{\Delta x_i \Delta y_j} \int_i dx \int_j dy \psi_n(x,y) \tag{44}$$

and

$$Q_{ij} = \frac{1}{\Delta x_i \Delta y} \int_i dx \int_j dy S(x,y). \tag{45}$$

Hence, dividing Eq. (39) by $\Delta x_i \Delta y_j$ we obtain the spatial balance equation

$$\frac{g_n \mu_n}{\Delta x_i} (\psi_{n,i+1/2,j} - \psi_{n,i-1/2,j}) + \frac{g_n \mu_n}{\Delta y_j} (\psi_{ni,j+1/2} - \psi_{ni,j-1/2}) + \Sigma_{ij} \psi_{nij} = Q_{ij}. \tag{46}$$

Equation (46) contains no spatial approximations. Along with this expression of neutron balance, we require two auxiliary relations to relate the cell average to the cell edge fluxes. The diamond difference approximation is[7]

$$\psi_{nij} = 1/2(\psi_{n,i+1/2,j} + \psi_{n,i-1/2,j}) \tag{47}$$

$$\psi_{nij} = 1/2(\psi_{ni,j+1/2} + \psi_{ni,j-1/2}). \tag{48}$$

For each case the diamond difference relationships are used as follows:

$$\psi_{n,i+1/2,j} = 2\psi_{nij} - \psi_{n,i-1/2,j} \quad ; \quad \mu_n > 0, \tag{49}$$

$$\psi_{n,i-1/2,j} = 2\psi_{nij} - \psi_{n,i+1/2,j} \quad ; \quad \mu_n < 0, \tag{50}$$

$$\psi_{ni,j+1/2} = 2\psi_{nij} - \psi_{ni,j-1/2} \quad ; \quad \eta_n > 0. \tag{51}$$

$$\psi_{ni,j-1/2} = 2\psi_{nij} - \psi_{ni,j+1/2} \quad ; \quad \eta_n < 0. \tag{52}$$

In each of the octants the appropriate pair of these equations is combined with Eq. (46) to obtain the necessary differenced equation. For the octant $\mu_n > 0$, $\eta_n > 0$. Equations (49) and (51) are used to eliminate $\psi_{n,i+1/2,j}$ and $\psi_{ni,j+1/2}$ from Eq. (46). We then have, upon solving for ψ_{nij},

$$\psi_{nij} = \left[\Sigma_{ij} + \frac{2\,g_n\mu_n}{\Delta x_i} + \frac{2\,g_n\eta_n}{\Delta y_j}\right]^{-1}\left[\frac{2\,g_n\mu_n}{\Delta x_i}\psi_{n,i-1/2,j} + \frac{2\,g_n\eta_n}{\Delta y_j}\psi_{ni,j-1/2} + S_{ij}\right]; \quad \mu_n > 0,\ \eta_n > 0. \tag{52}$$

Likewise, for the octant $\mu_n < 0$, $\eta_n > 0$ we obtain

$$\psi_{nij} = \left[\Sigma_{ij} + \frac{2\,g_n|\mu_n|}{\Delta x_i} + \frac{2\,g_n\eta_n}{\Delta y_j}\right]^{-1}\left[\frac{2\,g_n|\mu_n|}{\Delta x_i}\psi_{n,i+1/2,j} + \frac{2\,g_n\eta_n}{\Delta y_j}\psi_{n,i,j-1/2} + S_{ij}\right]; \quad \mu_n < 0,\ \eta_n > 0. \tag{53}$$

With these equations the procedure for upsweeping the grid shown in Fig. 4 may be explained in more detail.

Solution Algorithm

Performance of the upsweep constitutes a solution of Eq. (49) through (53), where the scattering source is written in the difference form

$$S_{ij} = \Sigma_{sij}\sum_{\eta_n>0} w_n\psi_{nij} + S_{+ij}. \tag{54}$$

The upsweep source S_{+ij} is considered known since, as indicated in Eq. (33), it consists of the group source and the ϕ_- (downward) component of the scalar flux, which was calculated on the preceding downsweep.

We consider first the case of vacuum boundary conditions at the lower boundary

$$\psi_{ni,1/2} = 0; \qquad \eta_n > 0 \tag{55}$$

and on the left and right boundaries:

$$\psi_{n,1/2,j} = 0; \qquad \eta_n > 0 \tag{56}$$

$$\psi_{n,I+1/2,j} = 0; \quad \mu_n < 0 \tag{57}$$

The upsweep proceeds by calculating the angular flux successively on the lines y_1, $y_{3/2}$, $y_2 \cdots y_{j-1/2}$, y_j, $y_{j+1/2} \cdots$. The values at $y_{1/2}$ are known from Eq. (55). Once the values along a line of cell centers (i.e. at y_j) have been calculated, the diamond difference relationship Eq. (51) is used to calculate the values at $y_{j+1/2}$ in terms of those at y_j and $y_{j-1/2}$. Thus we need concern ourselves only with the solutions of the pairs of Eqs. (49) and (52), and with (50) and (53), where $\psi_{ni,j-1/2}$ in these equations have already been determined.

Equations (52) through (54) are solved iteratively using an "inner-inner" or "line" iteration. The line iteration is initiated by guessing the first term on the right of Eq. (54) on the very first upsweep, and using the value calculated from the previous upsweep thereafter. After each line iteration the first term in Eq. (54) is updated until convergence of S_{ij} within some specified maximum pointwise error is obtained.

Each line iteration consists of the following. The angles are divided into pairs for which $\mu_{n'} = -\mu_n$ and $\eta_{n'} = \eta_n$, where $\eta_n > 0$ and $\mu_n > 0$. Starting at the left hand boundary, Eq. (56) is used in Eq. (52) to solve for ψ_{n1j}. Then Eqs. (49) and (52) are used alternately to march to the right solving successively for $\psi_{n,3/2,j}$ $\psi_{n2j} \cdots \psi_{n,I-1/2,j}$ $\psi_{nIj} \psi_{n,I+1/2}$. Similarly using Eq. (57) in Eq. (53) we may solve for $\psi_{n'I,j}$. Then Eq. (50) and (53) are used alternately to march to the left solving successively for $\psi_{n',I-1/2,j} \psi_{n',I-1,j} \cdots \psi_{n',3/2,j}$ $\psi_{n',1,j}$ $\psi_{n',1/2,j}$. This procedure is repeated for each of the $N(N+2)/8$ pairs of directions before the value of the first term or the right of Eq. (54) is updated.

Reflected boundary conditions are handled in a manner analogous to other discrete ordinate codes. If the reflected boundary is on the right the sweep to the left is initiated using the last value calculated in the sweep to the right:

$$\psi_{n',I+1/2,j} = \psi_{n,I+1/2,j}. \tag{58}$$

Conversely if the reflected boundary is on the left, the left (n') sweep is performed first, and the sweep to the right (n) is initiated with the condition

$$\psi_{n,1,j} = \psi_{n',1,j}. \tag{59}$$

If reflected boundaries are present on both right and left, the line iteration is initated using surface angular flux values saved from the previous upsweep. Thereafter, the surface angular flux from the previous line iteration is used.

If a reflected boundary appears on the lower surface (i.e. at $y_{1/2}$), the values of $\chi_{ni,1/2}$ calculated in the preceding downsweep are used to start the upsweep. If both upper and lower boundaries are reflected, then angular flux values from the previous outer iteration are used to initiate either upsweep or downsweep.

II.5. r-z Difference Equation

The discrete ordinate equations for ψ_n in r-z geometry are[6]

$$g_n \left[\frac{\mu}{\rho} \frac{\partial}{\partial \rho} \rho - \frac{1}{\rho} \frac{\partial}{\partial \omega} + \xi \frac{\partial}{\partial z} \right] \psi_n(\rho,z) + \Sigma(\rho,z)\psi_n(\rho,z) = S(\rho,z) \qquad (60)$$

$\hat{\Omega}_n$

where hereafter the radial (r) distance is denoted by ρ. The scattering source is given by

$$S(\rho,z) = \Sigma_s \sum_{\xi_n>0} w_n \psi_n(\rho,z) + S_+(\rho,z) \qquad (61)$$

In r-z geometry z is taken as the polar axis; hence the polar angle is $\cos^{-1}\xi$, and the aximuthal angle is measured from the positive ρ axis. Thus

$$\mu = (1 - \xi^2)^{1/2} \cos \omega; \qquad \eta = (1 - \xi^2)^{1/2} \sin \omega. \qquad (62)$$

For brevity we have written ψ and S in terms of $\vec{r}$ and $\hat{\Omega}$, realizing that in r-z geometry

$$\vec{r} = \vec{r}(\rho,z), \qquad (63)$$

and $\hat{\Omega}$ is expressed in terms either of μ or η or ξ and ω. The flux, and hence the emmision density, must be symmetric about $\omega = 0$:

$$\psi(\vec{r},\xi,\omega) = \psi(\vec{r},\xi,-\omega). \qquad (64)$$

In what follows we first difference the ω dependence of the streaming term and carry through the appropriate angular discretization. We then apply diamond differencing to the spatial variables to derive a solution algorithm.

Angular Discretization

For the discretization of the angular variables we choose a level symmetric quadrature set[7], where the directions are enumerated pairs of $\pm \mu_i$, $\pm \xi_j$ for i = 1 .....N/2. While full rotational symmetry is not required it is important to place the quadrature points on levels of constant η_j about the z axis, for only with such an arrangement is it possible to numerically approximate the derivative with respect to ω in Eq. (60) without the complication of differing values η_j entering into the expressions.

To facilitate the angular discretization we associate two indices with each direction: $\hat{\Omega}_n \rightarrow \hat{\Omega}_{pq}$. The first index indicates the value ξ_p associated with $\hat{\Omega}_n$ with $\xi_1 < \xi_2 < \xi_3 < \ldots < \xi_N$. The second index increases with the value of μ associated with ξ_p. Hence for fixed ξ_p, $\mu_{p1} < \mu_{p2} \ldots < \mu_{N_p}$ where N_p varies

between 1 and N. This numbering scheme is illustrated schematically in Fig. 5, where the ordinate points are projected on the μ,ξ plane. In the case where anisotropic transport is considered the angular multiplier, g_n is a function only of ξ_n. Thus $g_n \rightarrow g_p$.

We may now express the discrete ordinates approximation to Eq. (60) by treating the angular derivative by the widely used α coefficient technique[7]:

$$g_p \frac{\mu_{pq}}{\rho} \frac{\partial}{\partial \rho} \rho \psi_{pq}(\vec{r}) + g_p \left[\frac{\alpha^p_{q+1/2} \psi_{p,q+1/2}(\vec{r}) - \alpha^p_{q-1/2} \psi_{p,q-1/2}(\vec{r})}{\rho w_{pq}} \right] + g_p \xi_p \frac{\partial}{\partial z} \psi_{pq}(r) + \Sigma \psi_{pq}(\vec{r}) = S(\vec{r}), \tag{65}$$

where the scalar flux is given by

$$\phi = 1/4 \sum_{pq} w_{pq} \psi_{pq}(\vec{r}) \tag{66}$$

The differencing coefficients now carry a superscript p to indicate that they are different for each level of ξ_p at which the derivative with respect to ω is evaluated.

The criteria for determining the alphas are given elsewhere.[6] The results are

$$\alpha^p_{1/2} = 0, \tag{67}$$

$$\alpha^p_{q+1/2} = \alpha^p_{q-1/2} - \mu_{pq} w_{pq}, \qquad \begin{matrix} p = 1,2...N \\ q = 1,2...N_p \end{matrix} \tag{68}$$

where

$$\alpha^p_{1/2} = 0 \qquad p = 1,2...N. \tag{69}$$

Having determined the values of $\alpha^p_{q+1/2}$ we must now relate the half integer angular flux values $\psi_{p,q+1/2}$ to the values ψ_{pq}. We define the angular diamond difference approximation

$$\psi_{pq}(\vec{r}) = 1/2[\psi_{p,q+1/2}(\vec{r}) + \psi_{p,q-1/2}(\vec{r})] \qquad \begin{matrix} p = 1,2...N \\ p = 1,2...N_p. \end{matrix} \tag{70}$$

Solving this equation for $\psi_{p,q+1/2}$, we insert the result into Eq. (65) to eliminate $\psi_{p,q+1/2}$ and obtain

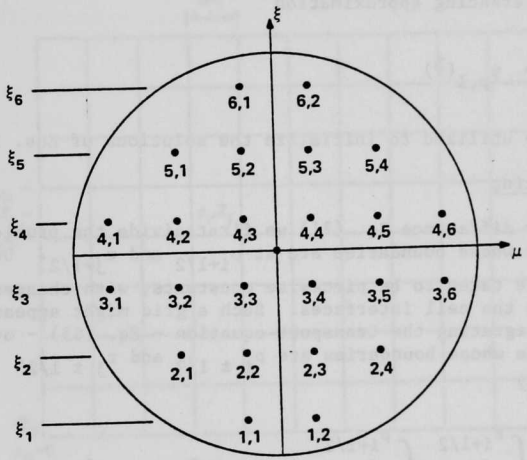


Fig. 5. Ordinate Numbering in r-z Geometry

$$g_p \frac{\mu_{pq}}{\rho} \frac{\partial}{\partial \rho} \rho \psi_{pq}(\vec{r}) + 2g_p \frac{\alpha^p_{q+1/2}}{\rho w_{pq}} \psi_{pq}(r) - g_p \frac{(\alpha^p_{q+1/2} + \alpha^p_{q-1/2})}{\rho w_{pq}} \psi_{p,q-1/2}(\vec{r})$$

$$+ g_p \xi_p \frac{\partial}{\partial z} \psi_{pq}(\vec{r}) + \sigma \psi_{pq}(\vec{r}) = q_{pq}(\vec{r}). \tag{71}$$

Once $\psi_{p,1/2}$ is known, Eqs. (70) and (71) may be solved for increasing q. Two techniques are available for treating $\psi_{p,1/2}$. In the starting direction method the transport is first solved in the directions $\mu = -1$ with $\xi = \xi_p$.

The alternative method used in DIF3D for initiating the calculation is to use the step differencing approximation

$$\psi_{p,1/2}(\vec{r}) = \psi_{p,1}(\vec{r}) \tag{72}$$

which then may be utilized to initialize the solutions of Eqs. (71) and (70).

Spatial Differencing

To spatially difference Eq. (71) we first divide the problems into an r-z grid of cells whose boundaries are at $\rho_{i+1/2}$ and $z_{j+1/2}$. Once again all cross sections are taken to be piecewise constants, with changes of value permitted only at the cell interfaces. Such a grid might appear as in Fig. 6. We proceed by integrating the transport equation - Eq. (65) - over an incremental volume whose boundaries are $\rho_{i \pm 1/2}$ and $z_{j \pm 1/2}$. Hence the volume is given by

$$V_{ij} = 2\pi \int_{z_{j-1/2}}^{z_{i+1/2}} dz \int_{\rho_{i-1/2}}^{\rho_{i+1/2}} d\rho \, \rho = \Delta z_j \pi (\rho^2_{i+1/2} - \rho^2_{i-1/2}). \tag{73}$$

We then apply the diamond difference approximation to the spatially discretized equations. Following the convention that

$$\int_{\rho_{i-1/2}}^{\rho_{i+1/2}} d\rho \equiv \int_i d\rho \quad \text{and} \quad \int_{z_{j+1/2}}^{z_{j-1/2}} dz \equiv \int_j dz, \tag{74}$$

the balance equation over the V_{ij} cell is then

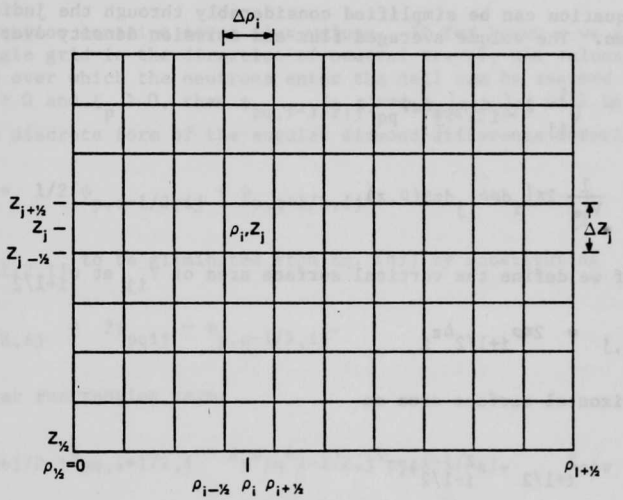


Fig. 6. Grid in r-z Geometry

$$g_p \mu_{pq} 2\pi \rho_{i+1/2} \int_j dz \psi_{pq}(\rho_{i+1/2}, z) - g_p \mu_{pq} 2\pi \rho_{i-1/2} \int_j dz \psi(\rho_{i-1/2}, z)$$

$$+ g_p \frac{\alpha^p_{q+1/2}}{w_{pq}} 2\pi \int_i d\rho \int_j dz \psi_{p,q+1/2}(\rho, z) - g_p \frac{\alpha^p_{q-1/2}}{w_{pq}} 2\pi \int_i d\rho \int_j dz \psi_{p,q-1/2}(\rho, z)$$

$$+ g_p \xi_p 2\pi [\int_i d\rho \rho \psi_{pq}(\rho, z_{j+1/2}) - \int_i d\rho \rho \psi_{pq}(\rho, z_{j-1/2})] + \Sigma_{ij} 2\pi \int_i d\rho \rho \int_j dz \psi_{pq}(\rho, z)$$

$$= 2\pi \int_i d\rho \rho \int_j dz S(\rho, z). \tag{75}$$

This equation can be simplified considerably through the judicious definition of terms. The volume averaged flux and emission density over V_{ij} are

$$\psi_{pqij} = \frac{1}{V_{ij}} 2\pi \int_i d\rho \rho \int_j dz \psi_{pq}(\rho, z), \tag{76}$$

$$S_{ij} = \frac{1}{V_{ij}} 2\pi \int_i d\rho \rho \int_j dz S(\rho, z). \tag{77}$$

Similarly if we define the vertical surface area of V_{ij} at $\rho_{i+1/2}$ as

$$A_{i+1/2,j} = 2\pi \rho_{i+1/2} \Delta z_i \tag{78}$$

and the horizontal surface area as

$$B_i = \pi(\rho^2_{i+1/2} - \rho^2_{i-1/2}), \tag{79}$$

we may write the surface averaged fluxes over these areas as

$$\psi_{pq,i+1/2,j} = \frac{2\pi \rho_{i+1/2}}{A_{i+1/2,j}} \int_j dz \psi_{pq}(\rho_{i+1/2}, z) \tag{80}$$

and

$$\psi_{pqi,j+1/2} = \frac{2\pi}{B_i} \int_i d\rho \rho \psi_{pq}(\rho, z_{j+1/2}). \tag{81}$$

Finally, we define

$$\psi_{p,q+1/2,ij} = \frac{2\pi}{A_{i+1/2} - A_{i-1/2}} \int_i d\rho \int_j dz \psi_{p,q+1/2}(\rho, z) \tag{82}$$

and rewrite Eq. (75) as

$$g_p\mu_{pq}A_{i+1/2,j}\psi_{pq,i+1/2,j} - g_p\mu_{pq}A_{i-1/2,j}\psi_{pq,i-1/2,j} + g_p\frac{\alpha^p_{q+1/2}}{w_{pq}} \times$$

$$(A_{i+1/2,j} - A_{i-1/2,j})\psi_{p,q+1/2,ij} - g_p\frac{\alpha^p_{q-1/2}}{w_{pq}}(A_{i+1/2,j} - A_{i-1/2,j})\psi_{p,q-1/2,ij}$$

$$+ g_p\xi_pB_i\psi_{pqi,j+1/2} - g_p\xi_pB_i\psi_{pqi,j-1/2} + V_{ij}\Sigma_{ij}\psi_{pqij} = V_{ij}S_{ij}. \qquad (83)$$

This equation contains seven flux values. If for known q we are to sweep the space-angle grid in the direction of neutral travel, the values of ψ on the surfaces over which the neutrons enter the cell can be assumed to be known. Thus if $\mu_{pq} > 0$ and $\xi_p > 0$, then $\psi_{pq,i-1/2,j}$ and $\psi_{pqi,j-1/2}$ will be known. Likewise the discrete form of the angular diamond difference formula,

$$\psi_{pqij} = 1/2(\psi_{p,q-1/2,ij} + \psi_{p,q+1/2,ij}), \qquad (84)$$

allows $\psi_{p,q+1/2,ij}$ to be eliminated from Eq. (83) by substituting

$$\psi_{p,q+1/2,ij} = 2\psi_{pqij} - \psi_{p,q-1/2,ij}. \qquad (85)$$

We have, after rearranging terms

$$g_p\mu_{pq}A_{i+1/2,j}\psi_{pq,i+1/2,j} - g_p\mu_{pq}A_{i-1/2,j}\psi_{pq,i-1/2,j}$$

$$\frac{2g_p\alpha^p_{q+1/2}}{w_{pq}}(A_{i+1/2,j} - A_{i-1/2,j})\psi_{pqij} - \frac{g_p(\alpha^p_{q+1/2} + \alpha^p_{q-1/2})}{w_{pq}}(A_{i+1/2,j} - A_{i-1/2,j}) \times$$

$$\psi_{p,q-1/2,ij} + g_p\xi_pB_i\psi_{pqi,j+1/2} - g_p\xi_pB_i\psi_{pqi,j-1/2} + V_{ij}\Sigma_{ij}\psi_{pqij} = V_{ij}S_{ij} \qquad (86)$$

Equations (85) and (86) are the spatially discretized forms of Eqs. (70) and (71). They are solved successively for increasing p. Before this can be done, however, the fluxes from the outgoing ρ and z surfaces of the V_{ij} cell must be eliminated from the equation. To do this we utilize the spatial diamond difference relationships:

$$\psi_{pqij} = 1/2(\psi_{pq,i+1/2,j} + \psi_{pq,i-1/2,j}) \tag{87}$$

$$\psi_{pqij} = 1/2(\psi_{pqi,j+1/2} + \psi_{pqi,j-1/2}) \tag{88}$$

As in other geometries these equations are used in different combinations for the four octants. Since for the upsweep we consider only the the case where $\xi_p > 0$, we rewrite Eq. (88) as

$$\psi_{pqi,j+1/2} = 2\psi_{pqij} - \psi_{pqi,j-1/2} \tag{89}$$

in order to eliminate $\psi_{pqi,j+1/2}$ from Eq. (86). Then for $\mu_{pq} < 0$ we write Eq. (87) as

$$\psi_{pq,i-1/2,j} = 2\psi_{pqij} - \psi_{pq,i+1/2,j}. \tag{90}$$

to also eliminate $\psi_{pq,i-1/2,j}$. Inserting these expressions into Eq. (86), we obtain

$$\psi_{pqij} = \{g_p|\mu_{pq}|(A_{i+1/2,j} + A_{i-1/2,j})\psi_{pq,i+1/2,j} + 2g_p|\xi_p|B_i\psi_{pqi,j-1/2,j}$$

$$+ w_{pq}^{-1}(A_{i+1/2,j} - A_{i-1/2,j})g_p(\alpha^p_{q+1/2} + \alpha^p_{q-1/2})\psi_{p,q-1/2,ij} + V_{ij}S_{ij}\}\cdot$$

$$\{g_p|\mu_{pq}|(A_{i+1/2,j} + A_{i-1/2,j}) + 2g_p|\xi_p|B_i + w_{pq}^{-1}(A_{i+1/2,j} - A_{i-1/2,j}) \times$$

$$(\alpha^p_{q+1/2} + \alpha^p_{q-1/2}) + \Sigma_{ij}V_{ij}\}^{-1}; \qquad \xi_p > 0, \ \mu_{pq} < 0. \tag{91}$$

Here Eq. (79) has been used to arrange the denominator so that the coefficients are the same as in the numerator.

Similarly, for $\mu_{pq} > 0$ we rewrite Eq. (87) as

$$\psi_{pq,i+1/2,j} = 2\psi_{pqij} - \psi_{pq,i-1/2,j} \tag{92}$$

and substitute this expression along with Eq. (89) into Eq. (86) to obtain

$$\psi_{pqij} = \{g_p\,\mu_{pq}\,(A_{i+1/2,j} + A_{i-1/2,j})\psi_{pq,i-1/2,j} + 2g_p\,\xi_p\,B_i\psi_{pqi,j-1/2,j}$$

$$+ w_{pq}^{-1}(A_{i+1/2,j} - A_{i-1/2,j})g_p(\alpha^p_{q+1/2} + \alpha^p_{q-1/2})\psi_{p,q-1/2,ij} + V_{ij}S_{ij}\} \times$$

$$\{g_p \mu_{pq} (A_{i+1/2,j} + A_{i-1/2,j}) + 2g_p \xi_p B_i + w_{pq}^{-1}(A_{i+1/2,j} - A_{i-1/2,j}) \times$$

$$(\alpha^p_{q+1/2} + \alpha^p_{q-1/2}) + \Sigma_{ij} V_{ij}\}^{-1}; \qquad \xi_p > 0, \ \mu_{pq} < 0. \tag{93}$$

Solution Algorithm

While the performance of the upsweep in r-z geometry follows the procedure in x-y geometry, the presence of the angular derivative term results in some additional effort. The scattering source term appearing in Eqs. (91) and (93) results from the discretization of Eq. (61):

$$S_{ij} = \Sigma_{sij} \sum_{\xi_p > 0} w_{pq} \psi_{pqij} + S_{+ij}. \tag{94}$$

The upsweep source S_{+ij} is considered known since, as indicated in Eq. (33), it consists of the group source and the ϕ_-(downward) component of the scalar flux, which are calculated on the preceeding downsweep.

Referring to Fig. 6, we consider here only the case of vacuum boundaries. Reflected boundaries on the top and/or bottom surfaces (i.e. at $z_{1/2}$ or $z_{J+1/2}$) are treated in a manner completely analogous to the ($y_{1/2}$ and $y_{J+1/2}$) surfaces in x-y geometry. The symmetry condition at $\rho_{1/2} \equiv 0$ is treated within the iterative scheme, while at the outer radius $\rho_{I+1/2}$, a vacuum condition is normally imposed. An option of the case also allows for isotropic return conditions at the outer radius.

Analogous to x-y geometry, the upsweep proceeds by calculating the angular flux successively on the lines z_1, $z_{3/2}$, $z_2 \cdots z_{j-1/2}$, z_j, $z_{j+1/2} \cdots$. The values as $z_{1/2}$ are known for the vacuum boundary condition

$$\psi_{pq,i,1/2} = 0. \tag{95}$$

Once the values along a line of cell centers (i.e. at z_j) have been determined, the diamond difference realtionship, Eq. (89) is used to calculate the values at $z_{j+1/2}$ in terms of those at z_j and $z_{j-1/2}$. Thus we need concern ourselves only with the solutions Eqs. (90) through (94), where the $\psi_{pqi,j-1/2}$ in these equations have already been determined. Equations (90) through (94) are solved iteratively using "inner-inner" or "line" iterations. Analogous to x-y geometry, the line iteration is initiated by guessing the first term on the right of Eq. (94).

II.6 Triangular Geometry Difference Equations

The starting point for the triangular geometry is the two-dimensional discrete ordinates equation in the x-y plane,

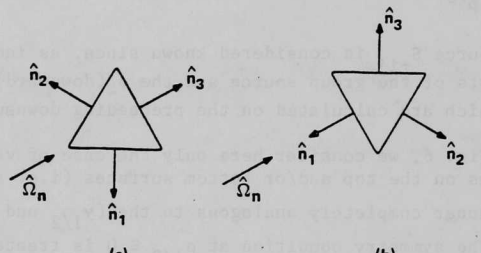


Fig. 7. Triangle Orientations

$$\hat{\Omega}_n \cdot \vec{\nabla}\psi_n(\vec{r}) + \Sigma(\vec{r})\psi_n(\vec{r}) = S(\vec{r}). \tag{96}$$

with piecewise constant cross sections, we require that the $\vec{r} = \vec{r}(x,y)$ domain of the problem be divided into a grid of equilateral triangles such that the cross sections change values only along the boundaries between triangles.

Spatial Discretization

Consider a particular equilateral triangle, with area A, side Length L, and outward normals to these sides of $\hat{n}_1$, $\hat{n}_2$ and $\hat{n}_3$ as indicated in Fig. 7. Integrating over the area of the triangle and applying the divergence theorem to the gradient term, we obtain

$$\sum_{i=1}^{3} \hat{\Omega}_n \cdot \hat{n}_i \int_i dL\psi_n(\vec{r}) + \Sigma \int_A dA\psi_n(\vec{r}) = \int_A dA S(\vec{r}), \tag{97}$$

where $\int_i$ indicates the integral over the length L of the side i. Defining area and edge averages as

$$\psi_n = \frac{1}{A} \int dA\psi_n(\vec{r}), \tag{98}$$

$$\bar{S} = \frac{1}{A} \int dA S(\vec{r}) \tag{99}$$

and

$$\psi_{ni} = \frac{1}{L} \int_i dL\psi_n(\vec{r}), \tag{100}$$

we may write Eq. (97) as

$$\sum_{i=1}^{3} \hat{\Omega}_n \cdot \hat{n}_i \frac{L}{A} \psi_{ni} + \Sigma\psi_n = \bar{S}. \tag{101}$$

This balance relationship is utilized to develop a spatial differencing scheme[11] that allows us to sweep through the triangular mesh in the directions of neutron travel. Viewing Fig. 7, however, it is immediately obvious that two different situations must be treated. In Fig. 7a, $\hat{\Omega}_n \cdot \hat{n}_1 < 0$, $\hat{\Omega}_n \cdot \hat{n}_2 < 0$. Hence ψ_{n1} and ψ_{n2} are fluxes of neutrons entering the cell and may be assumed to be known. Thus only one auxiliary relationship is needed to supplement Eq. (101) in determining ψ_n and ψ_{n3}. For this purpose, the following triangular generalization of the diamond difference formula is perfectly adequate.

$$\psi_n = \frac{1}{3} (\psi_{n1} + \psi_{n2} + \psi_{n3}). \tag{102}$$

In the orientation of Fig. 7b, however, only $\hat{\Omega}_n \cdot \hat{n}_1 < 0$, and the only known incoming flux is ψ_{n1}. Hence Eq. (101) and (102) are insufficient for determining ψ_n, ψ_{n2} and ψ_{n3}, and we must look for an additional relationship. This may be accomplished as follows.

Suppose that instead of dealing with only four flux values per triangle, as in the foregoing discussion, we deal with five values as indicated for the two orientations shown in Fig. 8. Thus for a triangle in which the neutrons enter across two sides, the five flux values consist of the average incoming flux and the fluxes at the two ends of the face across which neutrons leave. Conversely, for a triangle in which neutrons enter across only one face, we use the fluxes at the ends of that edge and the average flux values across the two edges through which neutrons leave the triangle. In each case two of the flux values are known, and when ψ_n is included three must be determined. Moreover since any triangular grid can be expressed by pairs of such triangles, with adjoining edges over which the corner flux must be determined, such a scheme will allow sweeping through the grid in the direction of neutron travel.

To proceed, we represent the angular flux distribution in a triangle by

$$\psi_n(x,y) \approx a_n + b_n x + c_n y. \tag{103}$$

With the origin of a local coordinate system at the centroid and the orientation shown in Fig. 9, we may write this interpolation on the average edge fluxes as

$$\psi_n(x,y) = \frac{\psi_{n1} + \psi_{n2} + \psi_{n3}}{3} + \frac{1}{L}(\psi_{n3} - \psi_{n2})x + \frac{1}{\sqrt{3}\,L}(2\psi_{n1} - \psi_{n2} - \psi_{n3})y. \tag{104}$$

By integrating over the triangle it is easily shown that Eqs. (101) and (102) remain valid. In addition, suppose we define the flux at the vertex opposite side i as $\tilde{\psi}_{ni}$. From Eq. (104) it follows that

$$\tilde{\psi}_{ni} = 3\psi_n - 2\psi_{ni}. \tag{105}$$

These auxiliary relationships taken together with Eqs. (101) and (102) allow us to determine the cell average fluxes along with the necessary outgoing flux values.

For the case shown in Fig. 8a, where the flux is entering across two edges, eliminating ψ_{n3} between Eqs. (101) and (102) yields

$$\psi_n = \frac{(|\hat{n}_3 \cdot \hat{\Omega}_n| + |\hat{n}_1 \cdot \hat{\Omega}_n|)\psi_{n1} + (|\hat{n}_3 \cdot \hat{\Omega}_n| + |\hat{n}_2 \cdot \hat{\Omega}_n|)\psi_{n2} + AS/L}{(3|\hat{n}_3 \cdot \hat{\Omega}_n| + \Sigma A/L)}. \tag{106}$$

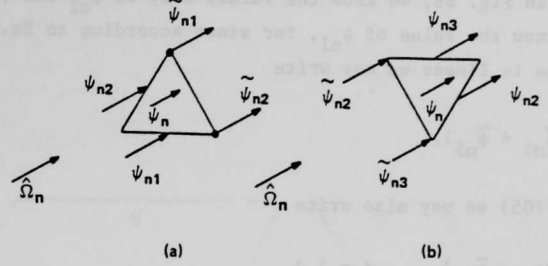


Fig. 8. Flux Nomenclature for Triangular Grid

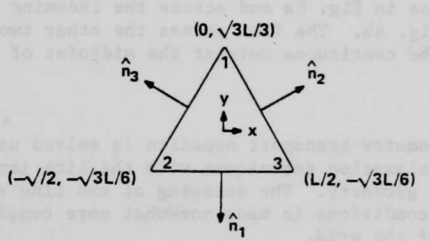


Fig. 9. Triangular Node Numbering

To obtain the outgoing flux values, $\tilde{\psi}_{n1}$ and $\tilde{\psi}_{n2}$, we simply use Eq. (105) with i = 1 and 2.

For the orientation in which the neutrons enter the triangle through only one face as shown in Fig. 8b, we know the values only of $\tilde{\psi}_{n2}$ and $\tilde{\psi}_{n3}$. Then, however, we also know the value of ψ_{n1}, for since according to Eq. (104) the flux along any edge is linear we may write

$$\psi_{n1} = 1/2(\tilde{\psi}_{n2} + \tilde{\psi}_{n3}). \tag{107}$$

By inverting Eq. (105) we may also write

$$\psi_{ni} = 1/2(3\psi_n - \tilde{\psi}_{ni}) \qquad i = 1,2. \tag{108}$$

Substituting thee three relationships into Eq. (101), we can solve for ψ_n in terms of $\tilde{\psi}_{n2}$ and $\tilde{\psi}_{n3}$:

$$\psi_n = \frac{(|\hat{n}_1\cdot\hat{\Omega}_n| + |\hat{n}_2\cdot\hat{\Omega}_n|)\tilde{\psi}_{n2} + (|\hat{n}_1\cdot\hat{\Omega}_n| + |\hat{n}_3\cdot\hat{\Omega}_n|)\tilde{\psi}_{n3} + 2AS_n/L}{(3|\hat{n}_3\cdot\hat{\Omega}_n| + 3|\hat{n}_3\cdot\hat{\Omega}_n| + 2\Sigma A/L)} \tag{109}$$

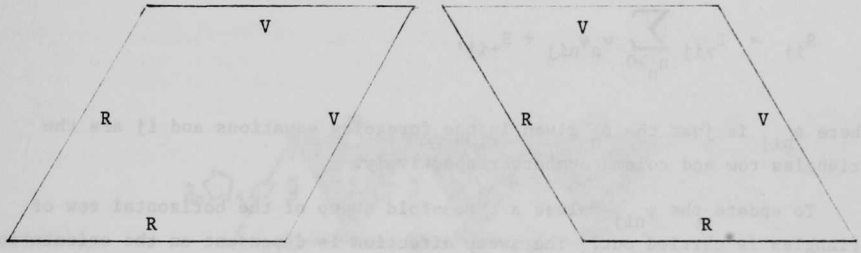
Thus this equation, along with Eq. (105), allows us to solve for the cell averaged and the outgoing flux values.

Note that while Eq. (104) specifies the flux as linear within the triangle, it says nothing about the continuity across the triangle's edges. Indeed, the flux is only forced to be continuous across the outgoing interface in triangles oriented as in Fig. 8a and across the incoming interface for those oriented as in Fig. 8b. The flux across the other two edges of the triangle is forced to be continuous only at the midpoint of the interface.

Solution Algorithms

The triangular geometry transport equation is solved using the same two-cyclic "up-down" overrelaxation techniques with the line inner iteration as applied in x-y and r-z geometry. The sweeping of the line and the treatment of reflected boundary conditions is made somewhat more complex, however, by the nonorthoganality of the grid.

Two geometries are considered: the 60 and 120 degree parallelogram shown in Fig. 10. Figure 10a and 10b normally correspond to 1/6 and 1/3 core calculations respectively, with reflected conditions on the bottom and left, and vacuum conditions on the top and right. Half and whole core configurations can also be treated by changing one or both of the reflected to vacuum conditions.



(a) 1/6 Core (b) 1/3 Core

Fig. 10. Triangular Grid Domains

To satisfy the reflected boundary conditions sixth fold symmetric quadrature sets from the DIAMANT2 code[10] are used. For a triangular S_N calculation there are a total of $3N(N+2)/4$ values of Ω_n; the projection of the 18 values for an S_4 calculation are projected onto the x-y plane in Fig. 11. Since we shall consider only the upsweep for illustrative purposes, we are concerned only with Ω_n n=1 through n=9 in the S_4 notation of Fig. 11.

To examine the upsweep consider first the configuration of Fig. 10a with all vacuum boundaries. The upsweep is carried out with one horizontal row at a time. Within each row a line inner-inner iteration is again carried out, updating the upward directed flux ϕ_+ after each line iteration until a specified level of convergence is reached. In triangular geometry the scattering source for the upsweep at a particular point is given by

$$S_{ij} = \Sigma_{xij} \sum_{\eta_n>0} w_n \psi_{nij} + S_{+ij},$$

where ψ_{nij} is just the ψ_n given in the foregoing equations and ij are the triangles row and column numbers respectively.

To update the ψ_{nij} values a three-fold sweep of the horizontal row of triangles is carried out. The sweep direction is dependent on the orientation of the neutron direction $\hat{\Omega}_n$ as indicated in Fig. 12. In each of the orientations the flux values along the lower horizontal line are known along with those along the right (or left) surface for which the neutrons enter. The sweep is carried out in the progression indicated by the numbers. In the unshaded triangles the incoming neutrons enter across two boundaries. Therefore Eq. (106) is used to calculate the flux at the triangles center in terms of those at the edge midpoints (indicated by parallelograms). Equation (105) is then used to calculate the flux at two of the triangles vertices (indicated by the solid circles). Since the flux is not continuous at the vertices across the boundaries on which only the midpoint fluxes are calculated, more than one node number is assigned per triangle vertex.

In the shaded triangles the neutrons enter through only one face. The flux value at the triangle center is calculated from Eq. (109) in terms of the vertex flux values lying in the triangle from which the neutrons came. Then Eq. (108) is used to calculate the flux values at the midpoints of the two faces through which the neutrons exit the triangle.

In situations where there are reflected boundary conditions along the lower and or left hand edge, the treatment is the same as in x-y geometry with a major exception. When a reflected boundary condition occurs on the left, some of the upsweep neutrons are reflected into the down sweep and vice versa. Thus the outgoing angular flux values must be saved from the preceeding up or down sweep.

The 1/3 core configuration shown in Fig. 10b is treated by internally redefining the problem in the same configuration as 10b. This is done by interchanging left and right boundary conditions in Fig. 10a and numbering the triangles from right to left.

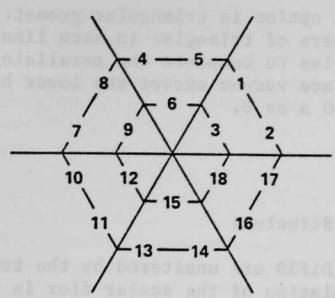


Fig. 11. Ordinate Array for Triangular Geometry

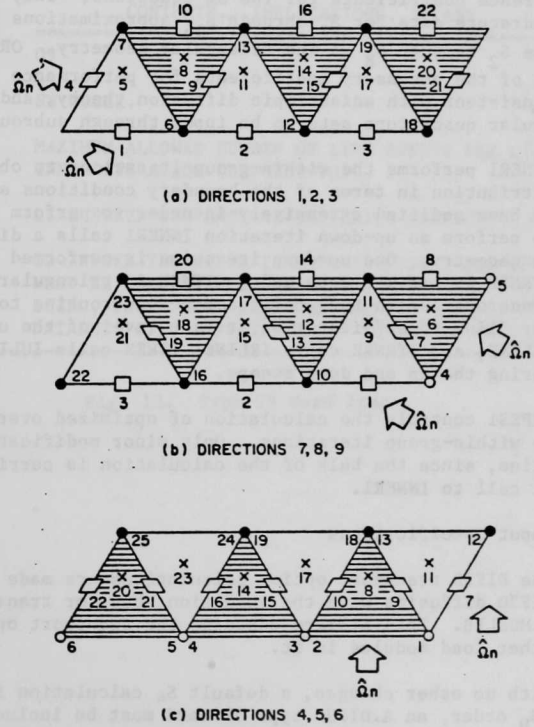


Fig. 12. Triangular Grid Sweeps

Consistent with the DIF3D diffusion option in triangular geometry the transport code can handle different numbers of triangles in each line so that it is not necessary to use filler triangles to complete the parallelogram. It is assumed that all boundary conditions are vacuum except the lower horizontal boundary and the left boundary in Fig. 10 a or b.

III. PROGRAM INFORMATION

III.1. Modifications of the DIF3D Structure

Most of the subroutines of DIF3D are unaltered by the transport option since only the within-group calculation of the scalar flux is changed. There are four modified subroutines: ORTFDC, TRIFDC, INNER1 and ORPES1.

ORTFDC and TRIFDC calculate the necessary difference coefficients to carry out the within-group scalar flux calculations for orthogonal and triangular geometries, respectively. These are changed to calculate the necessary difference coefficients for the S_N equations. They also include the angular quadrature data for S_2 through S_{16} approximations in x-y and r-z geometry and the S_2 through S_8 in the triangular geometry. ORTFDC includes the calculation of the necessary coefficients for performance of S_N calculations consistent with anisotropic diffusion theory, and an option allows user angular quadrature sets to be input through subroutine SNANGL.

INNER1 performs the within-group iterations to obtain the group scalar flux distribution in terms of the boundary conditions and group source. As such, it has been modified extensively in order to perform the up-down S_N iterations. To perform an up-down iteration INNER1 calls a different subroutine in each geometry. One up-down iteration is performed by ITRNXY in x-y geometry, by ITRNRZ in r-z geometry and by ITHEX in triangular geometry. Each of these three subroutines in turn calls another subroutine to perform the "inner-inner" or "line-inner" iteration at each level of the up or down sweep. ITRNXY calls IXLINE, and ITRNRZ calls IRLINE; ITHEX calls IULINE and IDLINE respectfully during the up and down sweeps.

ORPES1 controls the calculation of optimized overrelaxation factors for the within-group iterations. Only minor modification are required in this subroutine, since the bulk of the calculation is carried out in the repeated ORPES1 call to INNER1.

III.2. Input Specifications

The DIF3D transport option calculations are made using the same input as for DIF3D diffusion with the exception that for transport one "PRELIBs" to C116.B99983.MODLIB. This library contains the transport option of DIF3D; there are no other load modules in it.

With no other changes, a default S_4 calculation is provided. To change the S_n order, an A.DIF3D type 09 card must be included, See Fig. 13. This card also allows one to modify the convergence criteria and maximum number of sweeps for the "inner-inner" or "line-inner" iteration referred to in Section II.3.

```
C-------------------------------------------------------------------------
CR          SN TRANSPORT OPTIONS (TYPE 09)                              -
C                                                                       -
CL    FORMAT-----(I2,4X,2I6,6X,E12.4)                                   -
C                                                                       -
CD    COLUMNS          CONTENTS...IMPLICATIONS, IF ANY                  -
CD    =======  ========================================================-
CD    1-2      09                                                       -
CD                                                                      -
CD    7-12     SN ORDER.                                                -
CD                .                                                     -
CD    13-18    MAXIMUM ALLOWED NUMBER OF LINE SWEEPS PER LINE PER       -
CD             INNER ITERATION (DEFAULT=10).                            -
CD                                                                      -
CD    25-36    LINE SWEEP CONVERGENCE CRITERION (DEFAULT=1.0E-4).       -
C                                                                       -
CN             TO INVOKE THE DIF3D TRANSPORT OPTION, THE TYPE 09 CARD   -
CN             MUST BE PRESENT WITH A NONZERO SN ORDER. FOR THE TIME    -
CN             BEING, USERS MUST ALSO CONTINUE TO 'PRELIB' TO           -
CN             DATASET 'C116.B99983.MODLIB' TO INVOKE THIS OPTION.      -
C-------------------------------------------------------------------------
```

Fig. 13. Type 09 Card Image

III.3. Geometrical Restrictions

The transport option of DIF3D performs calculations only in two-dimensional x-y, r-z and triangular geometry. These correspond to geometries 40, 50 and 70 or 74 on the A.NIP3 type 03 card. Pseudo one-dimensional calculations in slab (or cylindrical) geometry can be performed as follows. Simply make the cross sections y (or z) independent in x-y (or r-z) geometry and utilize reflected boundary conditions on the upper and lower y (or z) boundaries. With one exception, the transport option of DIF3D will accept only zero flux and reflected boundary conditions from the standard DIF3D input file; zero flux boundary conditions are interpreted as vacuum boundary conditions in the S_N calculations. The exception is the isotropic return boundary condition which may be used on the outer radius of a r-z calculation. This is done by specifying a type 6 boundary condition on A.NIP3 type 04 card.

There are several restrictions on the combinations of boundary conditions that may be used in the transport options of DIF3D. In r-z geometry there must be either a zero flux (i.e. vacuum) or isotropic return boundary condition at the outer radius. The r-z boundary condition at $\rho = o$ is determined by the code from symmetry conditions, and therefore the input condition is ignored. In the two operational triangular geometries (70 and 74 on the A.NIP3 type 03 card) the right and top boundaries must be zero flux (i.e. vacuum).

It is important to be aware that in fully reflected systems the overrelaxation used to accelerate the within-group iterations becomes unstable. Therefore if all reflected or reflected and isotropic return boundary conditions are employed, one must use Gauss-Seidel iterations and expect slower convergence. This may be accomplished by inserting an overrelaxation factor of 1.00037 for each energy group on the A.DIF3D type 07 card.

III.4. Recalculations of Overrelaxation Factors

The cost incurred in calculating optimized overrelaxation factors may be comparable to several outer iterations. Thus it is sometimes advantageous to reuse overrelaxation factors from previous similar calculations, provided it is known that the factors do not change considerably from one calculation to another. To make such judgments the following rules of thumb may be helpful.

In general the S_N overrelaxation factors will be substantially smaller than those for the same problem run in the diffusion approximation. Thus it is inappropriate to use overrelaxation factors from a diffusion calculation in a transport calculation. Likewise it is often found to be computationally more efficient to use a larger error reduction factor on the within-group iterations convergence when S_N calculations are carried out. An increase in this factor by an order of magnitude over the diffusion default value on A.DIF3D type 06 card is recommended, for this often leads to increased computational efficiency.

The overrelaxation factors are quite insensitive to the S_N order and to the spatial grid spacing. Therefore the same overrelaxation factors may often be used while refining the space-angle discretization of the problem. The overrelaxation factors are somewhat more sensitive to the multi-group structure; generally they decrease as the group structure is refined.

III.5. Idiosyncrasies of S_N Codes

There are several pitfalls found in using S_N codes that do not appear in diffusion theory calculations. Some of these are generic to most or all S_N codes while others may be unique to the overrelaxation procedure used in the DIF3D transport option. Probably the two most important generic pitfalls relate to increasing the S_N order and to ray effects. More extensive discussions of these two problems may be found in standard texts on transport computational methods such as Ref. 6.

S_N ORDER REFINEMENT may not be carried out indefinitely without also refining the spatial mesh. Otherwise the truncation error in the space-angle differencing schemes becomes very large and negative scalar flux values or other nonphysical behavior may appear. The transport option of DIF3D, unlike some other S_N codes, has no negative flux fixup. Rather, the appearance of negative scalar flux values should serve as an indication that the spatial truncation errors may be significant. Conversely, it is always permissable to refine the spatial grid without increasing the S_N order. In effect such spatial refinement is a straightforward way to check whether unacceptable spatial truncation errors are present at a given S_N order.

RAY EFFECTS are anomalous wiggles that sometimes appear in the scalar flux distribution of S_N calculations. They are most likely to occur in situations where the group-source is localized and the ratio of within-group scattering to total cross sections is small. Therefore they tend to be a problem more in shielding than in core problems. Moreover their effects on integral parameters such as reactivity or region-averaged powers tend to be minimum. Nevertheless they have been known to occur. The greatest danger occurs when point flux values are used without making plots of the spatial distributions, for then it is likely that they will not be recognized.

If spatial flux plots reveal wiggly behaviors that is thought to be of suspicous origin, two tests may be applied to determine whether the oscillations are ray effects. First, a diffusion run may be carried out for the same problem; if the wiggles do not vanish completely, then they are real effects. Second, a higher order S_N calculation (with refined spatial mesh) may be carried out. If the wiggles decrease in magnitude and increase in frequency they are ray effects. If they are insensitive to the S_N order, they are real effects. In general there are no cheap and easy ways to eliminate ray effects; there does exist an extensive literature on the subject.[6]

III.6. Idiosyncrasies of DIF3D

On occasion difficulties may arise from the use of unaltered diffusion data files in the DIF3D transport option. These often are due to the nonphysical filler cross sections used in background regions, and in the thermal group of fast reactor calculations and to the group and region dependent transverse bucklings sometimes used in x-y and triangular calculations.

BACKGROUND REGION cross sections are often chosen arbitrarily or to increase the convergence rate of diffusion calculations. This often means that artificially larger total cross sections (i.e. small diffusion coefficients) are used. This however does not lead to increased convergence of the S_N calculation, and indeed the large spatial truncation errors may distort the solution near the boundaries. Convergence of S_N solution may be increased by maintaining a reasonable value of the total cross section (i.e. diffusion coefficient) but by decreasing or even eliminating the scattering cross section.

THERMAL GROUP calculations tend to converge very slowly in S_N calculations due to ratios of scattering to total cross section that often approach one. Thus if the thermal group does not effect the remainder of the calculation significantly, convergence can be accelerated by using an artificially small ratio of scattering to total cross section in the thermal group (no within-group thermal scattering is ideal).

TRANSVERSE BUCKLINGS derived from diffusion calculations tend to underestimate leakage, and in some cases may be negative in the lower energy groups. If the effects of such leakages are of large magnitude, they may create difficulties for the S_N difference equations, particularly when they are used in conjunction with artificial cross sections in the thermal energy group. One extreme case has been recorded where the thermal energy group in an S_N equation became unstable for this reason.

Acknowledgment

Several of the figures contained in this memorandum are copies of those provided by the Information Services Division of Los Alamos National Laboratory for the text E. E. Lewis and W. F. Miller, Jr., Computational Methods of Neutron Transport, Wiley, New York, 1984.

REFERENCES

1. D. R. Ferguson and K. L. Derstine, "Optimal Iteration Strategies and Data Management Considerations for Fast Reactor Finite Difference Diffusion Theory Codes," Nucl. Sci. Eng., 64, 593 (1977).

2. K. L. Derstine, "DIF3D: A Code to Solve One-, Two-, and Three-Dimensional Finite-Difference Diffusion Theory Problems," ANL-82-64, Argonne National Laboratory, 1983.

3. E. E. Lewis, "A Generalization of the Benoist Formalism for Anisotropic Neutron Transport Computation," Proc. ENS/ANS Topical Meeting on Advances in Mathematical Methods for the Solution of Nuclear Engineering Problems, Munich, April 27-29, 1981.

4. E. E. Lewis, "Anisotropic Transport in p-z Geometry," Trans. Am. Nucl. Soc., 39, 457 (1981).

5. L. A. Hageman, "Numerical Methods and Techniques Used in the Two-Dimensional Neutron Transport Program TPT," WAPD-TM-1125, Bettis Atomic Power Laboratory, 1973).

6. E. E. Lewis and W. F. Miller Jr., Computational Methods of Neutron Transport, Wiley, New York, 1984.

7. B. G. Carlson and K. D. Lathrop, "Transport Theory - The Methods of Discrete Ordinates," Computational Methods in Reactor Physics, H. Greenspan, C. N. Kelber and D. Okrent (eds.), Gorden and Breach, New York, 1968.

8. K. D. Lathrop and F. K. Brinkley, Jr., "TWOTRAN-II: An Interfaced, Exportable Version of the TWOTRAN Code for Two-Dimensional Transport, LA-4848-MS, Los Alamos National Laboratory, 1973.

9. R. J. Selva, "TPT01 Reference Manual," WAPD-TM-1120 (L), Bettis Atomic Power Laboratory, 1974.

10. K. Kufner and R. Hegen, "DIAMANT2, Ein Multiguppen Neutronentransport programm fur Dreiecks-und Hexagonalgeometrie," KFF3033, Kernforschung-zentrum, Karlsruhe, 1980.

11. Wm. H. Reed, "Triangular Mesh Difference Schemes for the Transport Equation," LA-4769, Los Alamos National Laboratory, 1971.

Appendix A

Table A-1
Discrete ordinates: weights and ordinates

S_2	w_n	μ_n	η_n
	1.0000000	0.5773503	0.5773503

S_4	w_n	μ_n	η_n
	0.3333333	0.9044490	0.3016388
	0.3333333	0.3016388	0.9044490
	0.3333333	0.3016388	0.3016388

S_6	w_n	μ_n	η_n
	0.1694466	0.9455768	0.2300919
	0.1638868	0.6881343	0.6881343
	0.1694466	0.2300919	0.9455768
	0.1638868	0.6881343	0.2300919
	0.1638868	0.2300919	0.6881343
	0.1694466	0.2300919	0.2300919

S_8	w_n	μ_n	η_n
	0.1167884	0.9622995	0.1923275
	0.0932552	0.7935218	0.5773503
	0.0932552	0.5773503	0.7935218
	0.1167884	0.1923275	0.9622995
	0.0932552	0.7935218	0.1923275
	0.0901032	0.5773503	0.5773503
	0.0932552	0.1923275	0.7935218
	0.0932552	0.5773503	0.1923275
	0.0932552	0.1923275	0.5773503
	0.1167884	0.1923275	0.1923275

S_{10}	w_n	μ_n	η_n
	0.0898420	0.9708020	0.1696223
	0.0672887	0.8450061	0.5071419
	0.0557801	0.6968602	0.6968602
	0.0672887	0.5071419	0.8450061
	0.0898420	0.1696223	0.9708020
	0.0672887	0.8450061	0.1696223
	0.0531338	0.6968602	0.5071419
	0.0531338	0.5071419	0.6968602
	0.0672887	0.1696223	0.8450061
	0.0557801	0.6968602	0.1696223
	0.0531338	0.5071419	0.5071419
	0.0557801	0.1696223	0.6968602
	0.0672887	0.5071419	0.1696223
	0.0672887	0.1696223	0.5071419
	0.0898420	0.1696223	0.1696223

S_{12}	w_n	μ_n	η_n
	0.0733218	0.9760093	0.1539575
	0.0526674	0.8756803	0.4576911
	0.0416149	0.7622583	0.6286966
	0.0416149	0.6286966	0.7622583
	0.0526674	0.4576911	0.8756803
	0.0733218	0.1539575	0.9760093
	0.0526674	0.8756803	0.1539575
	0.0389567	0.7622583	0.4576911
	0.0324902	0.6286966	0.6286966
	0.0389567	0.4576911	0.7622583
	0.0526674	0.1539575	0.8756803
	0.0416149	0.7622583	0.1539575
	0.0324902	0.6286966	0.4576911
	0.0324902	0.4576911	0.6286966
	0.0416149	0.1539575	0.7622583
	0.0416149	0.6286966	0.1539575
	0.0389567	0.4576911	0.4576911
	0.0416149	0.1539575	0.6286966
	0.0526674	0.4576911	0.1539575
	0.0526674	0.1539575	0.4576911
	0.0733218	0.1539575	0.1539575

S_{14}	w_n	μ_n	η_n
	0.0621716	0.9795154	0.1423897
	0.0433257	0.8960587	0.4204808
	0.0332176	0.8039850	0.5773503
	0.0318371	0.6999019	0.6999019
	0.0332176	0.5773503	0.8039850
	0.0433257	0.4204808	0.8960587
	0.0621716	0.1423897	0.9795154
	0.0433257	0.8960587	0.1423897
	0.0304863	0.8039850	0.4204808
	0.0245451	0.6999019	0.5773503
	0.0245451	0.5773503	0.6999019
	0.0304863	0.4204808	0.8039850
	0.0433257	0.1423897	0.8960587
	0.0332176	0.8039850	0.1423897
	0.0245451	0.6999019	0.4204808
	0.0199845	0.5773503	0.5773503
	0.0245451	0.4204808	0.6999019
	0.0332176	0.1423897	0.8039850
	0.0318371	0.6999019	0.1423897
	0.0245451	0.5773503	0.4204808
	0.0245451	0.4204808	0.5773503
	0.0318371	0.1423897	0.6999019
	0.0332176	0.5773503	0.1423897
	0.0304863	0.4204808	0.4204808
	0.0332176	0.1423897	0.5773503
	0.0433257	0.4204808	0.1423897
	0.0433257	0.1423897	0.4204808
	0.0621716	0.1423897	0.1423897

S_{16}	w_n	μ_n	η_n
	0.0541543	0.9820308	0.1334457
	0.0367965	0.9105818	0.3911943
	0.0277727	0.8330270	0.5368969
	0.0258028	0.7474682	0.6507561
	0.0258028	0.6507561	0.7474682
	0.0277727	0.5368969	0.8330270
	0.0367965	0.3911943	0.9105818
	0.0541543	0.1334457	0.9820308
	0.0367965	0.9105818	0.1334457
	0.0249427	0.8330270	0.3911943
	0.0196232	0.7474682	0.5368969
	0.0187976	0.6507561	0.6507561
	0.0196232	0.5368969	0.7474682
	0.0249427	0.3911943	0.8330270
	0.0367965	0.1334457	0.9105818
	0.0277727	0.8330270	0.1334457
	0.0196232	0.7474682	0.3911943
	0.0154480	0.6507561	0.5368969
	0.0154480	0.5368969	0.6507561
	0.0196232	0.3911943	0.7474682
	0.0277727	0.1334457	0.8330270
	0.0258028	0.7474682	0.1334457
	0.0187976	0.6507561	0.3911943
	0.0154480	0.5368969	0.5368969
	0.0187976	0.3911943	0.6507561
	0.0258028	0.1334457	0.7474682
	0.0258028	0.6507561	0.1334457
	0.0196232	0.5368969	0.3911943
	0.0196232	0.3911943	0.5368969
	0.0258028	0.1334457	0.6507561
	0.0277727	0.5368969	0.1334457
	0.0249427	0.3911943	0.3911943
	0.0277727	0.1334457	0.5368969
	0.0367965	0.3911943	0.1334457
	0.0367965	0.1334457	0.3911943
	0.0541543	0.1334457	0.1334457